Introduction

The workflow for variant calling using GATK is as follows -

- Variant Calling by running HalpotypeCaller on each sample (BAM file).
- Variant recalibration by running VQSR on the VCF resulted from joint genotyping.

HalpotypeCaller

The HaplotypeCaller is the main variant calling algorithm in GATK. It calls SNPs and indels simultaneously using local de novo assembly and a Bayesian statistical model. The HalpotypeCaller algorithm works in following steps :

Find Active Regions

First step in HalplotypeCaller procedure is to find the regions of high activity. These are the genomic regions which have strong evidence for variation. Each position is assigned a raw **activity score** which is the probability that the position contains a variant. This probability is calculated using the **reference-confidence model**. The raw profile thus obtained is then smoothered by copying the activity score over to the adjacent regions and then spreading out using a Gaussian kernel. Finally, the active regions are obtained as the ones conatining positions with high-enough activity score.
Mathematically, for each position i

- Assign score $P(i, variant)$ which is the probability of i being a variant.
- For each j in the interval around i of radius, $r \leq 50bp$, add $P(i, variant)$ to $P(j, variant)$. r is equal to the number of high quality soft-clipped bases that immediately follow or precede i.
- The profile score is spread out using Gaussian kernel upto 50bp. The score of i after spreading is denoted $score_i$.

The active regions are computed as follows -

- Final profile score, $Finalscore_i = \sum_k score_k$ where k runs over all the positions which contribute to the score of i as in previous step.
- Cut the genome at points where the final profile score goes from active to non-active regions by choosing a activity threshold.
- Trimm the not-so-important bps from the regions obtained in the previous step.

Local assembly of the active resgions

In this step the active regions are re-assembled de novo by building a de Bruijn type graph of the reference genome.

- Build a deBruijn graph (G) of the reference sequence.
- Initialize a weight hashtable (W) with edges as keys with weight 0.
- Build a hashtable ($UniqNodes$) of unique k-mers (nodes) of the graph with their positions in the graph.
- For each read, look for its k-mers in $UniqNodes$. If for a k-mer, there is a match and the $k-1$-mer is in W, increase the weight of the edge by 1. If a k-mer is not in the hashtable, we append it to the hashtable and add a new node in G.
- Prune the graph by throwing out the paths in the graph with weights below a threshold.

Identify Haplotypes

After the re-assembly graph is constructed and pruned, the halpotypes are extracted as -

- For each edge e in the graph, let i be the source vertex of e. Ccompute transition probability of e.

$$Prob(e) = W[e]/OutDegree(i)$$

  where $W[e]$ is the weight of the edge and $OutDegree(i)$ is the out degree of i.

- For each path p in the graph, compute its likelihood score as -

$$Likelihood(p) = \prod_{e \in p} Prob(e)$$

  The product is over all the edges in the path p.

- Select N paths with the highest likelihood scores. These are our potential Haplotypes.
- Align each halpotype to the reference genome and compute a CIGAR string for the haplotype.

Compute haplotype likelihoods

In the previous step, we computed a set of potential halpotypes. This set of sites gives a super-set of what will eventually be called-variants. The next step in GATK pipeline is to compute the evidence of existence of a haplotype given the data. For each haplotype H,

- Align each data read against H using **PairHMM** and compute the likelihood of observing the read given the halpotype. Thus we obtain $P(read|H)$ for all reads.
- Mariginalize the per-read likelihoods of halpotypes over alleles to get the per-read likelihoods for each allele, $P(read|A)$ for each allele A and read.
  For a given site, we list all the alleles observed in the data. Then, for each read, we look at the haplotypes that support each allele; we select the haplotype that has the highest likelihood for that read, and we write that likelihood in the new table. And that's it! For a given allele, the total likelihood will be the product of all the per-read likelihoods.
- Sites where one of the alleles has sufficient evidence will be the variants.

Bayesian Genotyper

GATK employed a Bayesian model to compute the most likely genotype of each sample at each site. The idea is to estimate the likelihoods of each possible genotype and predict the one with highest likelihood. The model is expressed by the following equation (Bayes theorem)

$$P(G|D) = \frac{P(G)P(D|G)}{\sum_i P(G_i)P(D|G_i)}$$

here $P(G)$ is the prior probability of the genotype G. GATK uses a flat prior.
Now the likelihood $P(D|G)$ can be expressed as

$$P(D|G) = \prod_j \left(\frac{P(D_j|H_1)}{2} + \frac{P(D_j|H_2)}{2} \right)$$

where $G = H_1H_2$ i.e. there are exactly two haplotypes (diploid assumption).
What remains to be figured out is $P(D_j|H_k)$, which is the per-read likelihood of the of the haplotype H_k aggregated over all the reads supporting the halpotype.This is exactly what we computed in the previous step.

- Compute $P(G_i|D)$ for all possible genotypes G_i.
- The predicted genotype is $argmax_{G_i}[P(G_i|D)]$

Appendices

Reference Confidence Model

The reference confidence model computes the probability of occurance of a variant at each position on the genome.

- Align the reads to the reference genome.
- At each position, estimate the probablility that some non-reference allele is segreegating at that position.
- The estimate of the probability that there is a variant at position i is given by

$$P(i, variant) = \text{Number of reads with reference base/Number of reads with non-reference base}$$